# 数值分析第二次上机练习实验报告

## ——非线性方程求根实验

力 4　杨昊光　2014011619

### 一、　问题的描述

用不同的方法求如下方程在$x_0 = 1$附近的根。（准确解为$x^* = 1.368808107\ldots$）

$$x^3 + 2x^2 + 10x - 20 = 0$$

如果方法收敛，希望达到 7 位有效数字：

(1) $x_{k+1} = \frac{20-2x_k^2-x_k^3}{10}$;

(2) $x_{k+1} = \sqrt[3]{20-10x_k-2x_k^2}$;

(3) 方法(1)的 Steffensen 加速方法；

(4) 方法(2)的 Steffensen 加速方法；

(5) Newton 法；

(6) 另外试着构造一种三阶收敛的格式。

### 二、　方法描述

1. 不动点迭代法及其收敛性分析

   (1)和(2)都有$\varphi(x^*) = x^*$，因而 $x^*$都是它们的不动点。

   (1) 对$x_{k+1} = \varphi(x_k) = \frac{20-2x_k^2-x_k^3}{10}$，取区间[1,2]作分析，则$\varphi$在[1,2]内单调递减，且

   $$\varphi(1) = 1.7\,;\ \varphi(2) = 0.4\,,$$

   $L$=1.3>1，即迭代函数在[1,2]内不收敛。

   (2) 对$x_{k+1} = \varphi(x_k) = \sqrt[3]{20-10x_k-2x_k^2}$，同样取[1,2]分析，得迭代函数单调递减，

   $$\varphi(1) = 2\,;\ \varphi(2) = -2\,,$$

   $L$=4>1，故迭代函数在[1,2]内也不收敛。

2. Steffensen 加速方法

   对迭代函数$x_{k+1} = \varphi(x_k)$进行 Steffensen 加速，得到新的迭代函数

   $$x_{k+1} = x_k - \frac{[\varphi(x_k) - x_k]^2}{\varphi(\varphi(x_k)) - 2\varphi(x_k) + x_k},$$

   代入(1)和(2)的迭代函数可以得到加速后的迭代函数，表达式如代码 5~10 行所述。

   对(1)和(2)，由于都有$\varphi'(x^*) \neq 0$ 且$\varphi'(x^*) \neq 1$，故 Steffensen 方法都是二阶收敛的。

3. Newton 法

   设$f(x) = x^3 + 2x^2 + 10x - 20$，有$f'(x) = 3x^2 + 4x + 10$，

   迭代函数为$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^3+2x_k^2+10x_k-20}{3x_k^2+4x_k+10}$。

4. 三阶收敛迭代函数的构造

   取$\gamma_j(x) = \mathcal{F}^{(j)}\big(f(x)\big)$，其中$\mathcal{F}(x)$为 $f(x)$的反函数，有：

   $$\gamma_1(x) = \frac{1}{f'(x)} = \frac{1}{3x^2+4x+10}\,, \gamma_2(x) = -\frac{f''(x)\gamma_1(x)}{[f'(x)]^2} = -\frac{6x+4}{(3x^2+4x+10)^3}.$$

   由此构造三阶收敛迭代函数$x_{k+1} = \varphi_3(x) = x_k - f(x_k)\gamma_1(x_k) + \frac{[-f(x_k)]^2}{2!}\gamma_2(x_k).$

## 三、 方案设计

编写 Matlab 程序实现迭代求近似根及其误差比较的过程。初始设定固定的迭代步数为 100 步，将初值和六种方法每一步迭代的结果保存进一个 101*6 的数组内。同时使用内置 roots()函数求解该方程的根作为参照值，用以计算出各步结果的剩余误差做分析。函数输入时为限定在实数范围内求解，立方根应使用函数 nthroot(F,3)，其中 F 为被开方的函数式。最后考察每一步迭代结果的收敛情况和有效位数。

## 四、 计算结果及其分析

要求的根为 $x^*$=1.368808107821373…

计算结果表明，迭代方法(1)和方法(2)不收敛：法(1)在 1.9232 到 0.5489 之间稳定振荡，法(2)在 3.1623 到-3.1623 之间稳定振荡；

方法(4)在第 7 次迭代后结果具有 7 位有效数字；

方法(3)和(5)在第 3 次迭代后结果达到 7 位有效数字；

方法(6)在第 2 次迭代后就已经有 7 位有效数字。

可见三阶收敛迭代方法具有显著优势。各方法的前 15 次迭代结果如下表 1 所示。对于收敛的四种方法，其迭代结果随迭代次数的变化折线画出如图 1。方法(3)~(6)的各次迭代后的误差也一并列出如表 2。

**表 1** 计算结果表

| 迭代次数 | 法(1) | 法(2) | Steffensen 加速后的法(1) | Steffensen 加速后的法(2) | Newton 法 | 三阶收敛迭代函数 |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1.70000000000000 | 2 | 1.33349213911386 | 1.20000000000000 | 1.41176470588235 | 1.36189700793812 |
| 2 | 0.930700000000000 | -2 | 1.36841543910839 | 1.27374039954537 | 1.36933647058824 | 1.36880806803185 |
| 3 | 1.74614203625570 | 3.17480210393640 | 1.36880805830916 | 1.30830003900789 | 1.36880818861753 | 1.36880810782137 |
| 4 | 0.857796793737152 | -3.17171550601145 | 1.36880810782137 | 1.34727521973943 | 1.36880810782137 | 1.36880810782137 |
| 5 | 1.78971892824351 | 3.16143810250219 | 1.36880810782137 | 1.36672391381434 | 1.36880810782137 | 1.36880810782137 |
| 6 | 0.786117463760370 | -3.16164373894025 | 1.36880810782137 | 1.36878928488277 | 1.36880810782137 | 1.36880810782137 |
| 7 | 1.82782332718813 | 3.16233360997041 | 1.36880810782137 | 1.36880810628964 | 1.36880810782137 | 1.36880810782137 |
| 8 | 0.721147914712977 | -3.16231990025473 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 9 | 1.85848552854863 | 3.16227393008962 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 10 | 0.667291268170302 | -3.16227484406792 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 11 | 1.88123148480484 | 3.16227790883989 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 12 | 0.626419796623837 | -3.16227784790799 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 13 | 1.89693882450978 | 3.16227764359028 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 14 | 0.597734533784599 | -3.16227764765240 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |
| 15 | 1.90718643311755 | 3.16227766127359 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 | 1.36880810782137 |

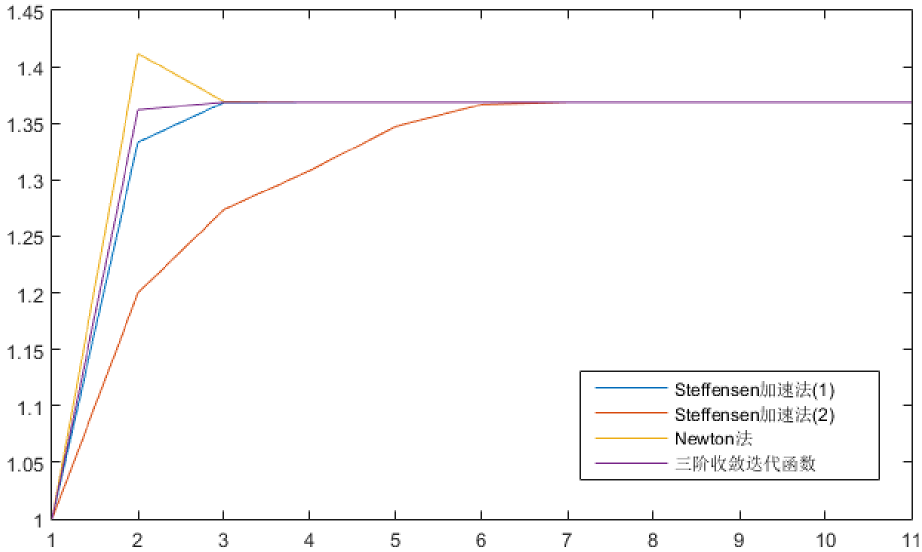
**表 2** 误差表

（2.220446e-16 为双精度数的机器 $\varepsilon$）

| 迭代次数 | Steffensen 加速法(1) | Steffensen 加速法(2) | Newton 法 | 三阶收敛迭代函数 |
|---|---|---|---|---|
| 0 | 0.368808107821373 | 0.368808107821373 | 0.368808107821373 | 0.368808107821373 |
| 1 | 0.0353159687075089 | 0.168808107821373 | 0.0429565980609803 | 0.00691109988324934 |
| 2 | 0.000392668712979161 | 0.0950677082759990 | 0.000528362766862633 | 3.97895254522496e-08 |
| 3 | 4.95122076671350e-08 | 0.0605080688134854 | 8.07961593185524e-08 | 2.22044604925031e-16 |
| 4 | 6.66133814775094e-16 | 0.0215328880819377 | 1.99840144432528e-15 | 2.22044604925031e-16 |
| 5 | 0 | 0.00208419400703708 | 2.22044604925031e-16 | 2.22044604925031e-16 |
| 6 | 2.22044604925031e-16 | 1.88229386053340e-05 | 2.22044604925031e-16 | 2.22044604925031e-16 |
| 7 | 2.22044604925031e-16 | 1.53173007610974e-09 | 2.22044604925031e-16 | 2.22044604925031e-16 |
| 8 | 2.22044604925031e-16 | 2.22044604925031e-16 | 2.22044604925031e-16 | 2.22044604925031e-16 |
| 9 | 2.22044604925031e-16 | 0 | 2.22044604925031e-16 | 2.22044604925031e-16 |
| 10 | 2.22044604925031e-16 | 2.22044604925031e-16 | 2.22044604925031e-16 | 2.22044604925031e-16 |

**图 1** 收敛折线图

（横轴为迭代次数+1）




## 五、 结论

迭代法是使用计算机进行非线性方程求根的常用方法，一般包括根所在区间的估计、迭代函数的构造、迭代求解三个步骤。其中，迭代函数的选择对迭代收敛性和收敛速度有决定性作用。一般而言，迭代函数阶数越高，迭代过程收敛越快。通过将待解方程移项分离出 $x$ 构造的迭代式和二分法一般是线性收敛的，经过 Steffensen 加速后的迭代式一般是至少二阶收敛的，牛顿法是典型的二阶收敛迭代法，经过求反函数的高阶导数构造的迭代式可以具有更高的收敛阶数。

然而，迭代法的收敛性一般是局部的，和函数在估计区间的不动点存在情况、导数情况和二阶导情况有关，需要具体问题具体分析。一阶迭代不收敛时，经过 Steffensen 加速等变换可以使其变得收敛，或使原来的收敛速度加快。比二阶更高阶的收敛函数在反函数求导方便时也可以使用。